

株式会社日本製鋼所 J-ADS プロバイダ

Version 1.0.1

ユーザーズ ガイド

December 27, 2022

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2019-06-12	初版.
1.0.1	2019-08-20	Controller クラス変数の変更, RawCommand コマンドの追加
	2022-12-27	ReadLGData1 コマンドに注釈を追加

【動作確認機種】

機種	バージョン	注意事項
SYSCOM5000i	-	

目次

1. はじめに.....	5
2. アプリケーション開発のための環境セットアップ.....	7
2.1. SYSCOM5000i とクライアント PC との接続.....	7
2.2. PC 開発環境のセットアップ.....	8
2.2.1. JSW J-ADS プロバイダのインストール.....	8
3. コマンドリファレンス.....	9
3.1. メソッド/プロパティ一覧.....	9
3.2. メソッド・プロパティ.....	9
3.2.1. CaoWorkspace クラス.....	9
3.2.1.1. AddController メソッド.....	9
3.2.2. CaoController クラス.....	10
3.2.2.1. VariableNames プロパティ.....	10
3.2.2.2. Variables プロパティ.....	11
3.2.2.3. AddVariable メソッド.....	11
3.2.2.4. Execute メソッド.....	11
3.2.3. CaoVariable クラス.....	14
3.2.3.1. Value プロパティ.....	14
3.3. 変数一覧.....	14
3.3.1. CaoController クラス変数.....	15
3.3.1.1. @MAKER_NAME.....	15
3.3.1.2. @VERSION.....	15
3.3.1.3. @LGDATA1.....	16
3.3.1.4. @ALARM_LOG.....	16
3.3.1.5. STATUS<??>.....	17
4. JSW J-ADS プロバイダによるプログラミング.....	18
4.1. 機械ステータスの値を取得するサンプルプログラミング.....	18
4.1.1. サンプルプログラム.....	19
4.1.1.1. 接続.....	20
4.1.1.2. 機械ステータスの値を取得.....	21

4.1.1.3. 切断	22
5. JSW J-ADS プロバイダエラーコード	23

1. はじめに

本書は、日本製鋼所（以下 JSW）社製 電動射出成形機デバイス J-ADS シリーズの ORiN プロバイダ（JSW J-ADS プロバイダ）のユーザーズガイドです。本プロバイダは JSW 社製 電動射出成形機 J-ADS から Ethernet 通信を利用してデータを取得します。図 1-1 が本プロバイダとデバイスの全体構成図になります。以降本プロバイダを JSW J-ADS プロバイダと呼称します。

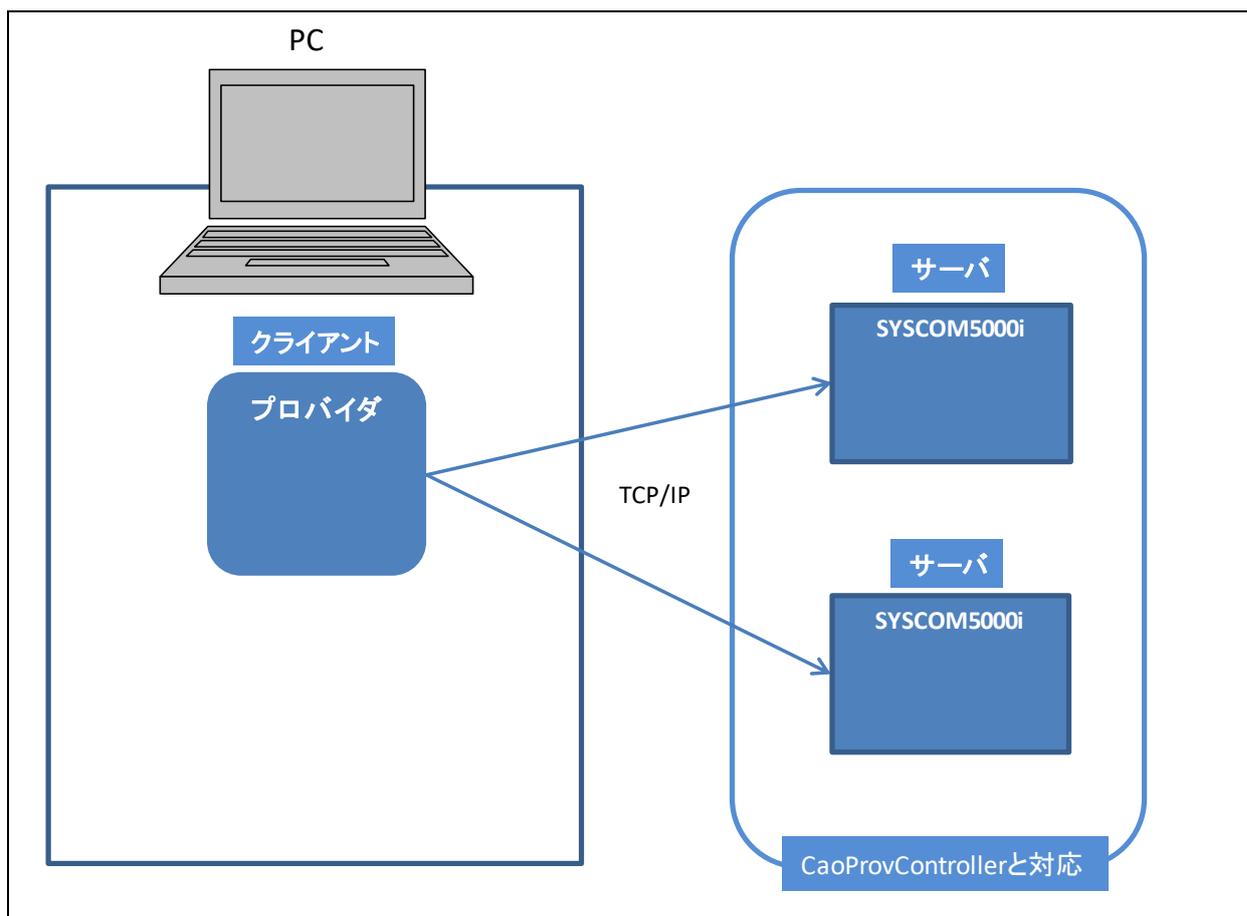


図 1-1 構成図

また、本プロバイダ及びデバイスそれぞれの対応を図 1-2に表します。
(※一例です。全てを表しているわけではありません。)

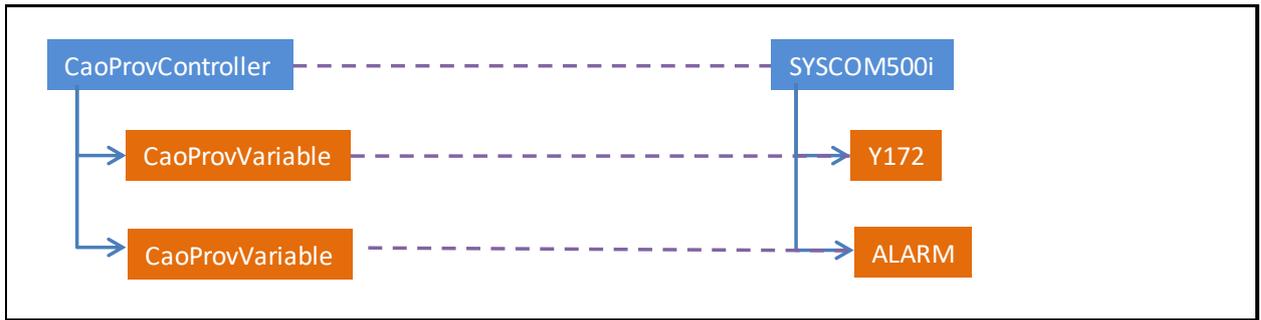


図 1-2 プロバイダの構成とデバイス情報との対応図

2. アプリケーション開発のための環境セットアップ

2.1. SYSCOM5000i とクライアント PC との接続

JSW J-ADS プロバイダは Ethernet 通信の TCP/IP 通信を用いて SYSCOM5000i と接続を行います。SYSCOM5000i の IP アドレスなどの設定は図 2-1 の画面で設定を行ってください。



図 2-1 SYSCOM5000i の設定画面

上記で設定した IP アドレスに接続できるように自身の PC の IP アドレスを設定し、機器との接続可能です。

2.2. PC 開発環境のセットアップ

2.2.1. JSW J-ADS プロバイダのインストール

JSW J-ADS プロバイダは ORiN2 SDK のインストーラにより自動的にセットアップされます。また、JSW J-ADS プロバイダを手動でインストールする場合は下記レジストリ登録を行う必要があります。レジストリ登録を行う場合は、管理者権限でコマンドプロンプトを起動し、regsvr32 コマンドを実行してください。

また、CAO エンジンが動作するには予め、PC 毎に正規の ORiN2 SDK ライセンスが1つ登録されていなくてはなりません。ORiN2 SDK ユーザーズガイド内にある「ライセンスの追加と削除」の節を参照してください。

表 2-1 JSW J-ADS プロバイダ

ファイル名	CaoProvJSWJ-ADS.dll
ProgID	CaoProv.JSW.J-ADS
レジストリ登録	regsvr32 CaoProvJSWJ-ADS.dll
レジストリ登録の抹消	regsvr32 /u CaoProvJSWJ-ADS.dll

3. コマンドリファレンス

3.1. メソッド/プロパティ一覧

表 3-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ ¹	機能	参照
CaoWorkspace			
	AddController	M コントローラに接続	P. 9
CaoController			
	VariableNames	P 接続可能な変数名リストの取得	P. 10
	Variables	P コントローラが保持する変数コレクションの取得	P. 11
	AddVariable	M 変数オブジェクトの追加	P. 11
	Execute	M 拡張コマンドの実行	P. 11
CaoVariable			
	Value	P 値の取得	P. 14

3.2. メソッド・プロパティ

3.2.1. CaoWorkspace クラス

3.2.1.1. AddController メソッド

CaoWorkspace に、コントローラオブジェクトを追加します。JSW J-ADS プロバイダでは、AddController メソッド実行時に渡されたパラメータを参照し、該当する SYSCOM5000i と接続を行います。以下に、AddController メソッドの仕様を示します。

書式

AddController

```
(
    "<コントローラ名>", // コントローラ名(任意)
    "CaoProv. JSW. J-ADS", // プロバイダ名(固定)
    "<マシン名>", // プロバイダ実行マシン名
    "<オプション>" // オプション文字列
)
```

オプション

以下にオプション文字列に指定するオプションを示します。オプション文字列は下記に示す各オプ

¹ M:メソッド, P:プロパティ, E:イベントをそれぞれ示します。

ションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲	デフォルト値
Conn=	○	接続用オプション (ETH)	-----	3.2.1.1.1 を参照
Timeout	--	応答待機時間 (ms)	1 - 30000	500

使用例

```
Dim engine As CaoEngine          ' Engineオブジェクト
Dim workspace As CaoWorkspace    ' Workspaceオブジェクト
Dim controller As CaoController  ' Controllerオブジェクト

Set engine = New CaoEngine
Set workspace = engine.Workspaces.Item(0)
Set controller = workspace.AddController("J-ADS", _
                                         "CaoProv. JSW. J-ADS", _
                                         "", _
                                         "Coon = ETH:192.168.0.1, Timeout = 1000")
```

3.2.1.1.1. Conn オプション

Conn = ETH:<接続先 IP[:<接続先ポート>[:<ローカル IP>[:<ローカルポート>]]]

<接続先 IP> : 接続先 IP アドレスを***.***.***.***の形式で指定します。

[<接続先ポート>] : 接続先ポート番号を指定します。

デフォルト: 55553 -> 複数クライアント対応

55554 -> 変数操作専用ポート

[<ローカル IP>] : PC 側の IP アドレスを***.***.***.***の形式で指定します。

デフォルト: 未指定

[<ローカルポート>]: PC 側のポート番号をしていします。

デフォルト: 未指定

3.2.2. CaoController クラス

3.2.2.1. VariableNames プロパティ

接続可能な変数名リストを取得します。本プロパティで取得した変数名は、後述する AddVariable メソッドの第一引数に使用することができます。

使用例

```
' 接続
Call Connect

' 変数名リスト取得
Dim variables as Variant
variables = controller.VariableNames
```

3.2.2.2. Variables プロパティ

コントローラが保持する、変数コレクションを取得します。

使用例

```
' 接続
Call Connect

' 変数コレクション取得
Dim variables as CaoVariables
Set variables = controller.Variables

' 変数取得
Dim variable as CaoVariable
Set variable = variables.Item(0)
```

3.2.2.3. AddVariable メソッド

CaoController に変数オブジェクトを追加します。変数名には3.3.1に示すもののみ使用できます。以下に、AddVariable の仕様を示します。

書式

AddVariable

```
(
    "<変数名>",           // 変数名
    "<オプション>"       // オプション文字列(省略可能)
)
```

3.2.2.4. Execute メソッド

ConController の拡張コマンドを実行します。以下に、Execute の仕様を示します。

書式

Execute

```
(
    "<拡張コマンド名>",   // 拡張コマンド名
    "<オプション文字列>"  // オプション文字列(省略可能)
)
```

以下に、Execute で指定できる拡張コマンド一覧を示します。使用例は拡張コマンドの詳細で記述しています。

コマンド	説明	参照
ReadStatus	指定した変数の値を取得します。	P. 12

コマンド	説明	参照
ReadLGDATA1	最新の測定値ログデータを取得します。	P. 13
ReadAlarmLog	現在のアラームを取得します。	P. 13
RawCommand	指定した任意のデータをそのまま送信し、機器とのやり取りを行います。	P. 13

3.2.2.4.1. ReadStatus コマンド

変数を指定し、指定した変数の値を取得します。以下に引数と戻り値を示します。ただし指定した変数が存在しない場合は、存在しない箇所のみ「?」が取得されます。

項目	型説明	
引数	VT_ARRAY VT_BSTR	
	/	VT_BSTR 指定する変数名を指定します。指定可能な変数は表 3-2 と表 3-3 をご参照ください。
戻り値	VT_ARRAY VT_BSTR	
	/	VT_BSTR 指定した変数の値が返却されます。

※ / : 指定する変数の個数

表 3-2 機械ステータスを表す変数一覧

変数名	概要	備考
Y172	機械状態（手動モード）	1（手動中） or 0（手動中でない）
Y175	機械状態（自動・半自動モード）	1（自動・半自動中） or 0（自動・半自動でない）
Y178	機械状態（休止中）	1（休止中） or 0（休止中でない）
Y179	機械状態（停止中）	1（停止中） or 0（停止中でない）
ALARM	シーケンサ（アラーム中）	1（アラームあり） or 0（アラームなし）

表 3-3 点検時期到達を表す変数一覧

変数名	概要	備考
INSM1NG	1ヶ月点検	1（点検時期到達） or 0
INSM3NG	3ヶ月点検	1（点検時期到達） or 0
INSM6NG	6ヶ月点検	1（点検時期到達） or 0
INSY1NG	1ヶ年点検	1（点検時期到達） or 0

使用例

```
' 接続
```

```
Call Connect
```

```
' 引数初期化
```

```
Dim param(1) As String
```

```
param(0) = "Y172"
```

```
param(1) = "ALARM"
```

```
' ReadStatus実行
```

```
Dim val() As String
```

```
val = controller.Execute("ReadStatus", param)
```

3.2.2.4.2. ReadLGDATA1 コマンド

測定値ログの最終の 1 レコードを取得します。以下に引数と戻り値を示します。

項目	データ型	型説明
引数	なし	
戻り値	VT_ARRAY VT_BSTR	測定値ログの最終の 1 レコードを要素とする配列が格納されます。 ²

使用例

```
' 接続
```

```
Call Connect
```

```
' ReadMeasurementLog実行
```

```
Dim val As Variant
```

```
val = controller.Execute("ReadLGDATA1", "")
```

3.2.2.4.3. ReadAlarmLog コマンド

未解除のアラーム情報を取得します。以下に引数と戻り値を示します。

項目	型説明
引数	なし
戻り値	VT_ARRAY VT_BSTR or VT_EMPTY

未解除アラーム情報が配列に格納されます。未解除アラームがない場合は、VT_EMPTY になります。

使用例

² レコード内の各要素の詳細は射出成形機ごとに異なります。メーカーへお問い合わせください。

```
' 接続
```

```
Call Connect
```

```
' ReadAlarmLog実行
```

```
Dim val As Variant
```

```
val = controller.Execute("ReadAlarmLog", "")
```

3.2.2.4.4. RawCommand コマンド

引数で指定された文字列を機器にそのまま送信し、機器からのデータをそのまま取得します。以下に引数と戻り値を示します。

項目	型説明	
引数	VT_BSTR	機器に送信する文字列を任意で指定します。
戻り値	VT_BSTR	機器からのデータをそのまま取得します。

※ 引数に不明なコマンドを指定した場合、下記のように戻り値が取得されます。

戻り値	VT_BSTR	UNKNOWN ○○○○
-----	---------	--------------

※○○○○には引数で指定した文字列が挿入されます。

使用例

```
' 接続
```

```
Call Connect
```

```
' RawCommand実行
```

```
Dim val As String
```

```
val = controller.Execute("RawCommand", "VGET Y172")
```

3.2.3. CaoVariable クラス

3.2.3.1. Value プロパティ

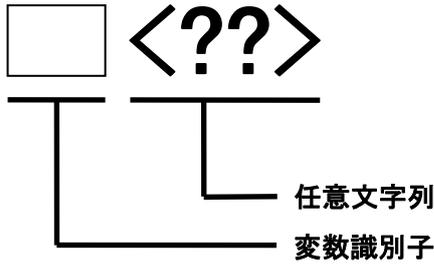
接続した SYSCOM5000i からデータを取得します。変数名によって動作が異なります。詳細は、3.3. 変数一覧を参照してください。

3.3. 変数一覧

各クラスで使用可能な変数一覧を定義します。なお変数は、CaoVariable クラスのオブジェクトを指します。複数変数を登録（オプションのみ変更したい場合等に有用）するために任意の文字列を付与することが可能です。

変数名に任意文字列を付与するための書式を以下に示します。

複数変数共通指定書式



3.3.1. CaoController クラス変数

変数名	説明	Value		参照
		get	put	
@MAKER_NAME	メーカー名を取得します.	○	-	P. 15
@VERSION	DLL バージョンを取得します.	○	-	P. 15
@LGDATA1	最新の測定値ログを取得します.	○	-	P. 16
@ALARM_LOG	未解除のアラームを取得します.	○	-	P. 16
STATUS<??>	変数の値を取得します.	○	-	P. 17

3.3.1.1. @MAKER_NAME

メーカー名の取得をします.

データ型

型説明	
VT_BSTR	メーカー名を取得します.

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@MAKER_NAME")
' 値取得
Dim strVal As String
strVal = var.value
```

3.3.1.2. @VERSION

DLL のバージョンの取得をします.

データ型

型説明

VT_BSTR	DLL のバージョンを取得します。 *. *. *
---------	------------------------------

使用例

```
' 接続
Call Connect
' 変数追加
Dim var As CaoVariable
Set var = controller.AddVariable("@VERSION")
' 値取得
Dim value As String
value = var.value
```

3.3.1.3. @LGDATA1

測定値ログの最終の 1 レコードを取得します。

データ型

型説明	
VT_ARRAY VT_BSTR	ReadLGDATA1 コマンドの戻り値と同様のデータが返却される。

使用例

```
' 接続
Call Connect
' 変数追加
Dim varLGDATA1 As CaoVariable
Set varLGDATA1 = controller.AddVariable("@LGDATA1")

' 値取得
Dim logValues() As String
logValues = varLGDATA1.value
```

3.3.1.4. @ALARM_LOG

未解除のアラーム情報を取得します。

データ型

型説明	
VT_ARRAY VT_BSTR or VT_EMPTY	ReadAlarmLog コマンドの戻り値と同様のデータが返却される。

使用例

```
' 接続
```

Call Connect

' 変数追加

```
Dim varAlarmLog As CaoVariable
Set varAlarmLog = controller.AddVariable("@ALARM_LOG")
```

' 値取得

```
Dim logValues() As String
logValues = varAlarmLog.value
```

3.3.1.5. STATUS<??>

以下のオプションで指定した変数の値を取得します。「STATUS」の後に任意の文字列を入力して変数名を指定してください。以下にオプション文字列で指定するオプションを示します。

オプション

オプション	必須	説明	値範囲	デフォルト値
Target=	○	取得したい変数名を指定します。 変数名をカンマ区切りで指定してください。なお、オプションの値は必ず以下のいずれかの括弧で囲ってください。 "()", "{}", "[]", "<>"	-----	-

データ型

型説明

VT_ARRAY VT_BSTR	ReadStatus コマンドの戻り値と同様のデータが返却される。
--------------------	-----------------------------------

使用例

' 接続

```
Call Connect
```

' 変数追加

```
Dim varStatus As CaoVariable
Set varStatus = controller.AddVariable("STATUS_1", "Target=<Y172, ALARM>")
```

' 値取得

```
Dim value() As String
value = varStatus.value
```

4. JSW J-ADS プロバイダによるプログラミング

JSW J-ADS プロバイダでは、以下の手順でクライアント PC と SYSCOM5000i を接続することができます。

- CaoEngine の作成
- CaoWorkspace の作成
- CaoController の作成

SYSCOM5000i に接続した後は、CaoController の Execute メソッドを使用する、もしくは、CaoVariable オブジェクトを生成することで、SYSCOM5000i の情報にアクセスすることができます。

4.1. 機械ステータスの値を取得するサンプルプログラミング

ここでは例としての機械ステータスの値を読み取るサンプルプログラムを示します。表 4-1 にサンプルプログラムの要件を、図 4-1 にサンプルプログラムの流れをそれぞれ記述しています。

表 4-1 サンプルプログラムの要件

要件	説明
接続先	TCP/IP で接続する
	接続先 IP アドレスは 192.168.1.2
	接続先ポート番号は 55553

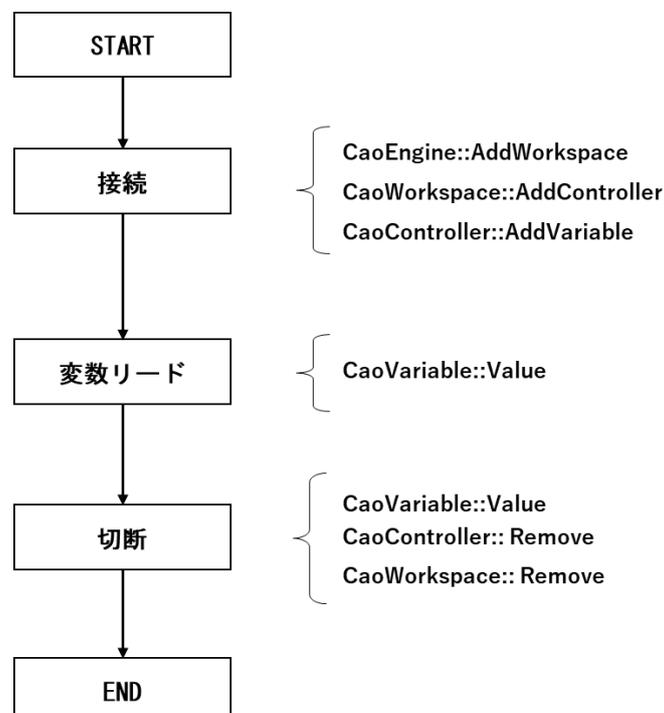


図 4-1 機械ステータスを表す変数の値を取得する流れ

以降の節から具体的なコードを示します。

4.1.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します。

Sample	ReadStatus.vb
--------	---------------

```
' 変数宣言
Private engine As CaoEngine
Private workspace As CaoWorkspace
Private controller As CaoController

' 接続メソッド
Private Sub Connect()
    ' CaoEngine オブジェクトの生成
    Set engine = New CaoEngine
    ' デフォルトの CaoWorkspace オブジェクトの取得
    Set workspace = engine.Workspaces.Item(0)

    ' CaoController オブジェクトの作成
    Set controller = workspace.AddController("J-ADS", _
                                           "GaoProv. JSW. J-ADS", _
                                           "", _
                                           "conn = ETH:192.168.1.2")
End Sub

' 切断メソッド
Private Sub Disconnect()
    workspace.Controllers.Remove(controller.Index)
    ' CaoController の消去
    Set controller = Nothing
    Call engine.Workspaces.Remove(workspace.Index)
    ' CaoWorkspace の消去
    Set workspace = Nothing
    ' CaoEngine の消去
    Set engine = Nothing
End Sub

Sub Main()
    ' 接続
    Call Connect
```

' 変数追加

```
Dim varStatus As CaoVariable
```

' CaoVariable オブジェクトの生成

```
Set varStatus = controller.AddVariable("STATUS_1", "Target=<Y172, ALARM>")
```

' 値取得

```
Dim statusValues() As String
```

```
statusValues = varStatus.value
```

' 返却値 1

```
Dim varY172 As String
```

```
varY172 = statusValues(0)
```

' 返却値 2

```
Dim varALARM As String
```

```
varALARM = statusValues(1)
```

```
Call controller.Variables.Remove(varStatus.Index)
```

```
Set varStatus = Nothing
```

' 切断

```
Call Disconnect
```

End Sub

4.1.1.1. 接続

SYSCOM5000i と接続するためには、以下の手順を取ります。

- (1) オブジェクトを保持するための変数を用意します。コントローラ接続に必要なオブジェクトは、CaoEngineオブジェクトとCaoWorkspaceオブジェクトとCaoControllerオブジェクトです。
CaoWorkspaceオブジェクトは、CaoControllerオブジェクトをCaoWorkspacesから取得する場合には変数を用意する必要はありません。また変数にアクセスするためのCaoVariableオブジェクトも必要になります。以下にVB6でのコード例を示します。

```
Private engine As CaoEngine
```

' CaoEngine オブジェクト用の変数

```
Private workspace As CaoWorkspace
```

' CaoWorkspace オブジェクト用の変数

Private controller As CaoController ' CaoController オブジェクト用の変数

- (2) CaoEngineオブジェクトを生成します。CaoEngineオブジェクトはNewキーワードを使って生成します。

```
' CaoEngine オブジェクトの生成
Set engine = New CaoEngine
```

- (3) CaoWorkspaceオブジェクトを取得もしくは生成します。CaoEngineオブジェクトを生成すると、デフォルトでCaoWorkspacesオブジェクトとCaoWorkspaceオブジェクトを1つずつ生成しています。以下にCaoWorkspaceオブジェクトを新しく取得するコード例とデフォルトのCaoWorkspaceを示します。

```
' デフォルトの CaoWorkspace オブジェクトの取得
Set workspace = engine.Workspaces.Item(0)
```

- (4) CaoControllerオブジェクトを生成します。CaoControllerオブジェクトを生成するには、使用するプロバイダ名と使用するためのパラメータを設定します。JSW J-ADSプロバイダでは、接続先情報をオプションで指定します。以下にコード例を示します。

```
' CaoController オブジェクトの生成
Set controller = workspace.AddController("J-ADS", _
                                         "CaoProv. JSW. J-ADS", _
                                         "", _
                                         "conn = ETH:192.168.1.2")
```

- (5) CaoVariableオブジェクトを生成します。接続したい変数のCaoVariableオブジェクトを生成します。以下に機械ステータスを表す変数のY172とALARMを指定し、データを取得する変数オブジェクトを生成するコード例を示します。

```
' 変数追加
Dim varStatus As CaoVariable
' CaoVariable オブジェクトの生成
Set varStatus = controller.AddVariable("STATUS_1", "Target=<Y172, ALARM>")
```

4.1.1.2. 機械ステータスの値を取得

値を取得するには、CaoVariable オブジェクトの Value プロパティを参照します。STATUS<??>変数の Value プロパティは VT_ARRAY | VT_BSTR 型のデータとなります。詳細は 3.3 を参照してください。以下にコード例を示します。

```
' 値を取得
Dim statusValues() As String
statusValues = varStatus.value
```

```
' 返却値 1
```

```
Dim varY172 As String  
varY172 = statusValues(0)
```

```
' 返却値 2
```

```
Dim varALARM As String  
varALARM = statusValues(1)
```

4.1.1.3. 切断

コントローラと切断する場合には、生成したオブジェクトを消去すると共に、オブジェクトを管理するコレクションクラスから消去するオブジェクトを削除します。以下にコード例を示します。

```
' CaoWorkspace から CaoController を削除  
Call workspace.Controllers.Remove(controller.Index)  
' CaoController の消去  
Set controller = Nothing  
' CaoEngine から CaoWorkspace を削除  
Call engine.Workspaces.Remove(workspace.Index)  
' CaoWorkspace の消去  
Set workspace = Nothing  
' CaoEngine の消去  
Set engine = Nothing
```

5. JSW J-ADS プロバイダエラーコード

本プロバイダには、独自のエラーコードは存在しません。

ORiN2 の共通エラーについては、「[ORiN2 プログラミングガイド](#)」のエラーコードの章を参照してください。

また、本プロバイダは、SYSCOM5000i から返却される [NG ○○] (○○は付録 B のエラーコード) のエラーコードを「0x8010****」でマスクして返します。また、不明なコマンドを機器に送った場合に、機器から返却される UNKNOWN レスポンスは RawCommand コマンド同様に、戻り値として文字列で取得されます。

SYSCOM5000i から返却されるエラーコードは付録 B を参照してください。

付録A. 通信プロトコルコマンド対応表

CaoController::Execute

コマンド	通信コマンド
ReadStatus	VGET
ReadLGDATA1	LGET LGDATA1
ReadAlarmLog	LGET ALARM_LOG

CaoVariable::Value

変数名	通信コマンド
@LGDATA1	LGET LGDATA1 -1
@ALARM_LOG	LGET ALARM_LOG -1
STATUS<??>	VGET

付録B. SYSCOM5000i からのエラーコード一覧

エラー番号	説明
08	指定した変数が存在しない.
09	指定した変数が管理変数ではない.